PB88-163019

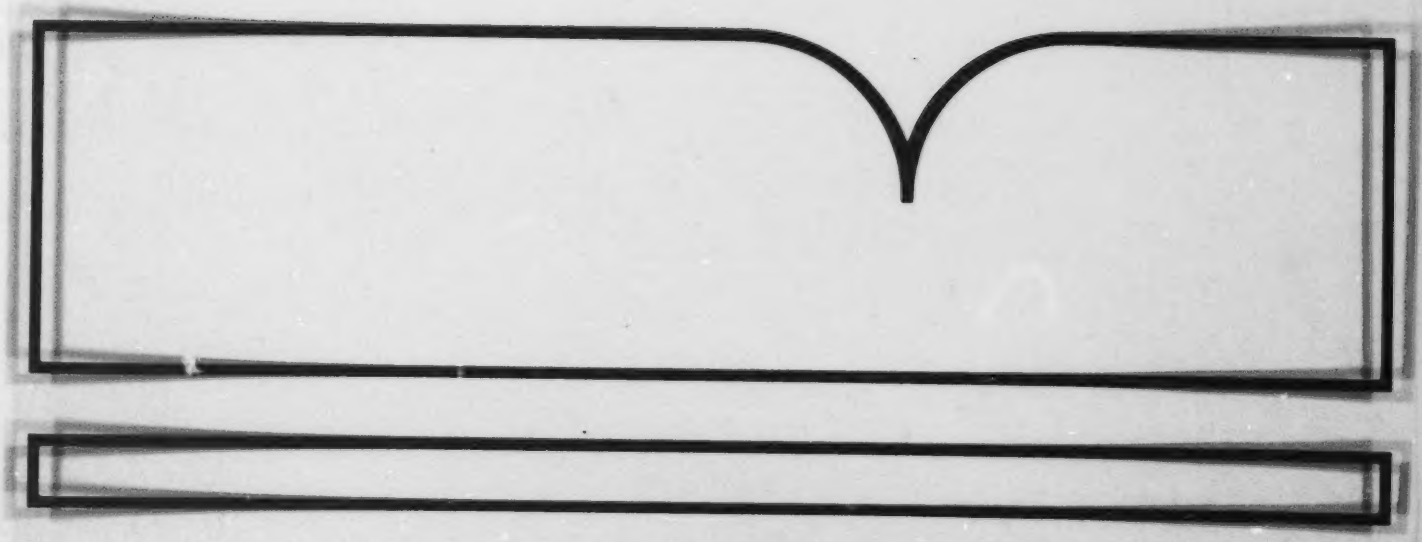
Toxic Air Pollutant/Source Crosswalk Information Storage and Retrieval System User's Manual

Pacific Environmental Services, Inc., Durham, NC

Prepared for

Environmental Protection Agency
Research Triangle Park, NC

Dec 87



U.S. Department of Commerce
National Technical Information Service

PB88-163019

United States
Environmental Protection
Agency

Office of Air Quality
Planning and Standards
Research Triangle Park NC 27711

EPA-450/4-87-023b
December 1987

Air

EPA

TOXIC AIR POLLUTANT/ SOURCE CROSSWALK

Information Storage And Retrieval System User's Manual

REPRODUCED BY
U.S. DEPARTMENT OF COMMERCE
NATIONAL TECHNICAL
INFORMATION SERVICE
SPRINGFIELD, VA. 22161

EPA-450 4-87-023b
December 1987

TOXIC AIR POLLUTANT/SOURCE CROSSWALK

Information Storage And Retrieval System User's Manual

By

Noncriteria Pollutant Programs Branch

U.S. ENVIRONMENTAL PROTECTION AGENCY
Office Of Air And Radiation
Office Of Air Quality Planning And Standards
Research Triangle Park, North Carolina 27711

This report has been reviewed by the Office Of Air Quality Planning And Standards, U.S. Environmental Protection Agency, and approved for publication. Any mention of trade names or commercial products is not intended to constitute endorsement or recommendation for use.

EPA-450/4-87-023b

TECHNICAL REPORT DATA
(Please read Instructions on the reverse before completing)

1. REPORT NO.	2.	3. RECIPIENT'S ACCESSION NO.
EPA-450/4-87-023b		PB88.163 019

4. TITLE AND SUBTITLE	5. REPORT DATE
Toxic Air Pollutant/Source Crosswalk - Information Storage and Retrieval System User's Manual	December 1987
	6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S)	8. PERFORMING ORGANIZATION REPORT NO.
9. PERFORMING ORGANIZATION NAME AND ADDRESS Pacific Environmental Services, Inc. 1905 Chapel Hill Road Durham, NC 27707	10. PROGRAM ELEMENT NO.
	11. CONTRACT/GRANT NO. 68-02-3887
12. SPONSORING AGENCY NAME AND ADDRESS U.S. Environmental Protection Agency OAR, OAQPS, AQMD, PCS (MD-15) Research Triangle Park, NC 27711	13. TYPE OF REPORT AND PERIOD COVERED Final
	14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

EPA Project Officer: Anne A. Pope

16. ABSTRACT

This report presents instructions on the Toxic Air Pollutant Crosswalk System that was developed to allow easy access and updates to the crosswalk data presented in the accompanying EPA report, Toxic Air Pollutant/Source Crosswalk - A Screening Tool For Locating Possible Source Emitting Toxic Air Pollutants, EPA-450/4-87-023a. This software system is an IBM PC based application that contains a listing of industrial source categories (SIC Codes), emitting source classifications (SCC codes), and their likely associated pollutants. The crosswalk program has modules that allow users to add, delete, edit, or browse specific data and to print tables contained in the accompanying crosswalk document. EPA is making available the system and data and data base records to agency users on three high-density diskettes.

The purpose of this report is to document the usage of the system. The system was designed as a tool to develop the report, Toxic Air Pollutant/Source Crosswalk - A Screening Tool For Locating Possible Sources Emitting Toxic Air Pollutant.

17. KEY WORDS AND DOCUMENT ANALYSIS

a. DESCRIPTORS	b. IDENTIFIERS/OPEN ENDED TERMS	c. COSATI Field/Group
Air Toxics Toxic Emissions Data System Standard Industrial Classification (SIC) Source Classification (SCC)		

18. DISTRIBUTION STATEMENT	19. SECURITY CLASS (This Report)	21. NO. OF PAGES
Unlimited	Unclassified	61
	20. SECURITY CLASS (This page)	22. PRICE
	Unclassified	A04 14.95

EPA Form 2220-1 (Rev. 4-77) PREVIOUS EDITION IS OBSOLETE

TABLE OF CONTENTS

Section		Page
	List of Figures	iv
1.0	Introduction	1
2.0	Getting Started	3
3.0	Notes On Printer Operation	5
4.0	Enter Data	7
5.0	Search/Browse/Edit Data	13
6.0	Synchronization Utility	19
7.0	Delete Records	21
8.0	Reports	23
9.0	Back Up Files	25
10.0	Notes On Index Files	27
11.0	Notes On Compiling The Source Code	29
Appendix A.	Source Code Listings	

LIST OF FIGURES

Figure	Title	Page
1	Main Menu	4
2	SIC/SCC/Pollutant Entry Screen	8
3	Pollutant Update Screen	9
4	Pollutant Update Screen (Pre-Edit Mode)	10
5	Search Screen	14
6	Browse Screen	16
7	Edit Screen	17
8	Synchronize SIC/SCC and Pollutant Data	20
9	Reports Menu	24
10	Index Files	28

1.0 INTRODUCTION

The Standard Industrial Classification (SIC) - Source Category Classification (SCC) / Pollutant Crosswalk System is an IBM PC based application designed to allow easy access and updates to information. This system, which contains the relationships of SIC and SCC codes and their associated pollutants, is designed for EPA and other users to access and/or update information in the accompanying report, Toxic Air Pollutant/Source Crosswalk - A Screening Tool For Locating Possible Sources Emitting Toxic Air Substances, EPA-450/4-87-023a.

The system consists of several modules - data entry and editing, data searches, and reports. The reports module of the Crosswalk program prints the tables contained in the accompanying crosswalk report. These tables present the related data from several different viewpoints. The tables include pollutant, SIC/SCC/Pollutant, SIC/Pollutant, and SCC/Pollutant listings.

A module exists for entering SIC/SCC/Pollutant data or updating pollutant listings. Another module allows the entry of search criteria to find either a specific SIC/SCC/Pollutant association, or a range of SIC/SCC/Pollutant relationships. These SIC/SCC/Pollutant data can be browsed and edited from tailored screens or listed to a printer.

This report presents instruction on the Air Toxics Pollutant Crosswalk System that was developed in the course of a Work Assignment that updated crosswalk data. The accompanying EPA report presents this updated crosswalk information. This manual exists to instruct EPA and other agencies on how to access the crosswalk system. The report was the primary product of the work assignment. The system was designed to update the information and is not the primary product of this work assignment. EPA has chosen to distribute copies of the system on a limited basis to assist those agencies who want to update and access the crosswalk data presented in the crosswalk report. This documentation discusses the detailed usage of each part of the system as well as other useful tips on the operation of the computer and dBase III PLUS ®.

2.0 GETTING STARTED

The Information Storage and Retrieval System is written in dBase III PLUS® (Ashton Tate) and compiled using the QuickSilver ®Compiler (WordTech Systems). The files necessary to operate the Information Storage and Retrieval System are the following:

PROGRAM FILES	DATABASE FILES		INDEX FILES	OTHER FILES
XWALK.EXE	SIC_SCC.DBF	---->	SIC_SCC.NDX	XWALK.BAT
		\->	SCC_POLL.NDX	ERROR.TXT
		\->	POLL_SIC.NDX	
	POLL.DBF	---->	POLL.NDX	
		\->	CAS.NDX	
		\->	CAS1.NDX	

An IBM PC or compatible computer with at least 512 Kilobytes of Random Access Memory (RAM) and a hard disk with approximately 4 Megabytes of free storage capacity is necessary for successful execution of the program. The files are distributed on diskettes created with the DOS BACKUP command. To install the files from the backup diskettes, see the instructions on how to Restore files in Section 8.0 BACKUP UP FILES.

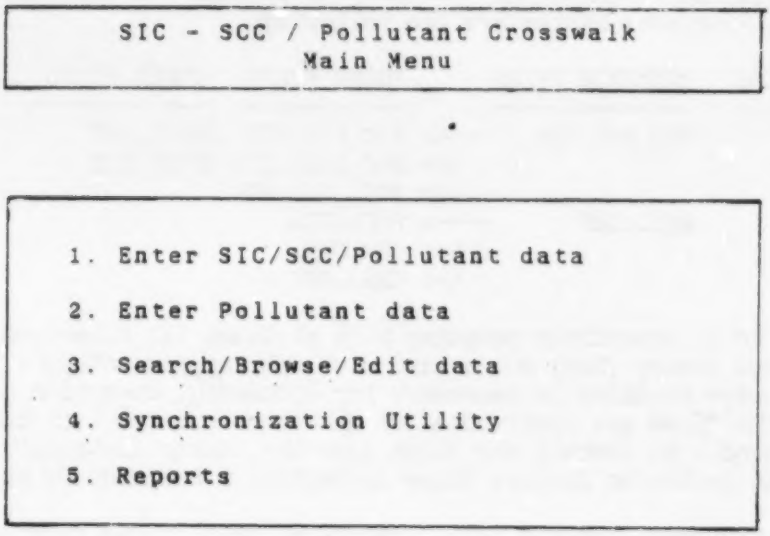
All of the above program (EXE), database (DBF), index (NDX), and text (TXT) files must be located in a directory named XWALK on a hard disk. The directory can be created by entering the DOS command MKDIR XWALK at the hard disk prompt. The XWALK.BAT file must be in the root directory of the hard disk or in a directory identified with the DOS PATH command. The DOS SET command is used to display the current path. The PATH is normally set in the AUTOEXEC.BAT file located in the root directory of the hard disk.

Since the dBase III PLUS® source code is compiled, the dBase III PLUS ® interpreter is not needed to execute the program. However, the source code has been provided on the distribution diskettes; these files have the extension PRG and FMT. A listing of the source code is also provided in Appendix A. These files do not need to be present to run the program.

To start the program type XWALK at the DOS prompt. Following the opening screen, the Main Menu will appear as shown in Figure 1. The different operations that can be performed are listed in the middle of the screen with the valid selection choices at the bottom of the screen. The valid selections are either numbers or the letters of the selection choices which are capitalized. For example, the selection choices in Figure 1 are "1 - 5 or Quit" and the valid entries would be 1, 2, 3, 4, 5 or Q.

Preceding page blank

FIGURE 1. MAIN MENU



3.0 NOTES ON PRINTER OPERATION

The Search/Browse/Edit data and Reports sections from the Main Menu can print reports to a printer. In order to prevent a program error caused by printer problems, please make sure the printer is on line and has plenty of paper before selecting any reports.

The user will be prompted to enter printer setup codes to achieve desired print quality the first time any reports are selected. These settings are saved to a file called PRT.MEM and used each time printing is done. If a new or different printer is used, simply erase the PRT.MEM file and the program will prompt you for new codes. These decimal settings can be found in the Control Codes section of your printer's manual.

During printing of any reports or retrievals, the F10 key may be pressed to abort the report generation and the program will return to the menu from which the report was selected.

4.0 ENTER DATA

Selection 1 from the Main Menu will give you a screen as shown in Figure 2. This selection allows the user to add SIC/SCC/Pollutant Crosswalk data. This screen updates the information contained in Tables 2-3 through 2-5 in the accompanying report.

In entering the data, the pollutant must be entered exactly as shown in the pollutant tables. Identical pollutant names are necessary because searches for CAS numbers are executed from the Pollutant database. When a match is found, the corresponding CAS number for that pollutant is automatically entered into the CAS Number field. If no match is found, the CAS Number field remains blank indicating the entered pollutant is not in the Pollutant database. This will alert users to the fact that either the pollutant spelling or nomenclature is incorrect or that the pollutant needs to be entered into the Pollutant database (Selection 2 from the Main Menu, described below).

Selections upon completion of a screen include "Add more" which adds the current screen's data to the database and produces a blank screen for more data entry, "End/save" which saves the current screen's data and returns to the main menu, "Quit/nosave" which returns to the main menu without saving the current screen's data, and "Redo screen" which places the cursor at the first field on the screen. These instructions may be selected by entering A, E, Q, or R, respectively.

Selection 2 from the Main Menu will produce a screen as shown in Figure 3. This screen is used for updating the pollutant data in Tables 2-1 and 2-2 contained in the accompanying report. Pollutants are entered and a search for the CAS number is performed from the Pollutant database. If a match is found, the corresponding CAS number will automatically be entered into the CAS Number field. The user will then be in the pre-edit mode and will be presented with a screen as shown in Figure 4. The selections at this point include "Delete" for deleting the screen's data from the pollutant database, " Edit" for editing the current screen's data, "Retry" for the ability to change the pollutant, or "Quit" which returns to the Main Menu. The valid entries would be D, E, R, or Q respectively. Deleted records are not physically deleted until the user returns to the Main Menu, but are noted with a message at the top of the screen.

Upon selecting "Edit", the user may edit the data and save it to the database. The old entry is overwritten with either "Add/edit more" which saves the current screen's data and produces another update screen or "End/save" which saves the current screen's data and exits to the Main Menu.

Preceding page blank

FIGURE 2. SIC/SCC/POLLUTANT ENTRY SCREEN

```
SIC - SCC / Pollutant Crosswalk
SIC/SCC/Pollutant Update Screen
```

```
SIC:

SCC:    -  -   -

Pollutant:
CAS Number:
```

Add more, End/save, Quit/nosave or Redo screen:

FIGURE 3. POLLUTANT UPDATE SCREEN

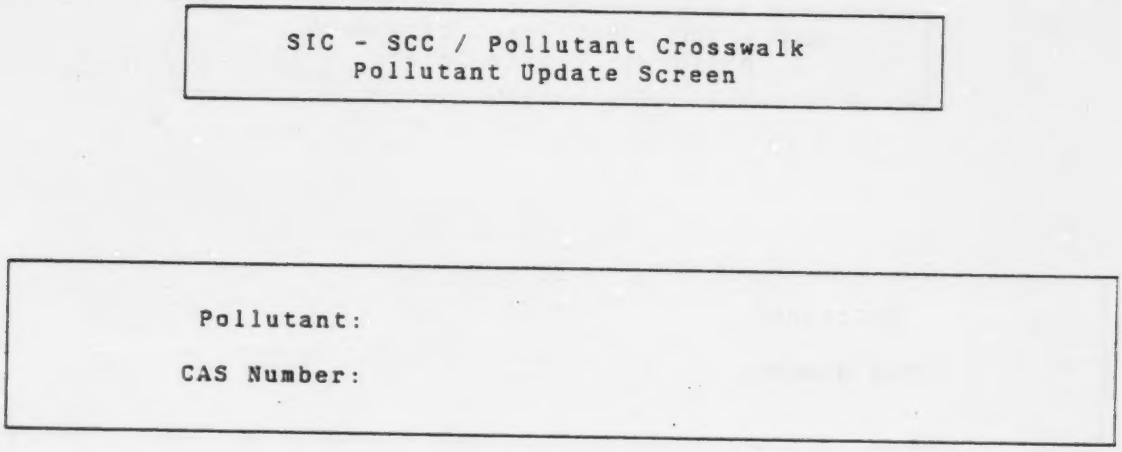
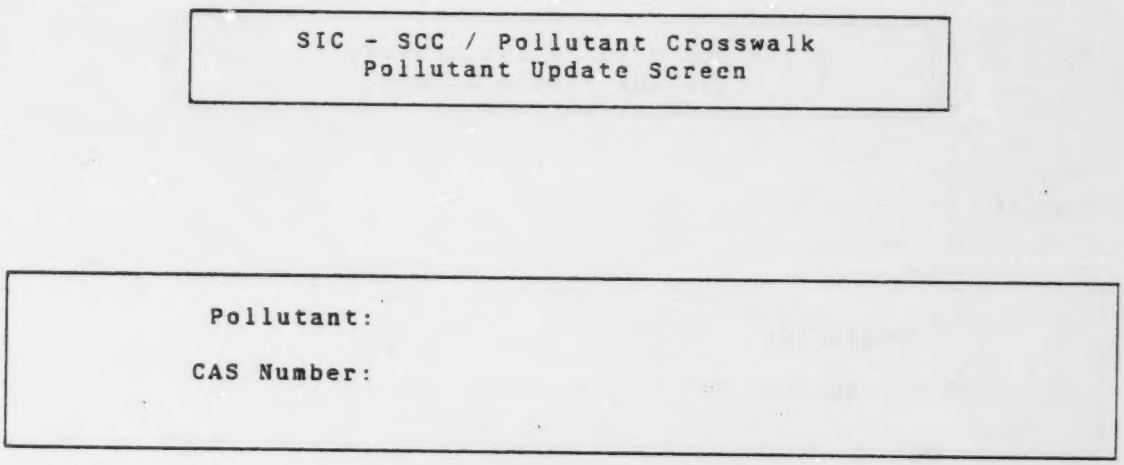


FIGURE 4. POLLUTANT UPDATE SCREEN (Pre-edit Mode)



When entering a pollutant into the Pollutant Update screen and a CAS number does not appear, the user may enter one and the data can be appended to the database. The valid screen selections are the same as in entering the SIC/SCC/POLLUTANT data. Data is saved only when either "Add/edit more" or "End/save" is selected.

On these data entry screens the numeric pad cursor keys, insert and delete are active and function as long as the Num Lock key is turned off. The Backspace key (left arrow <-) moves the cursor backward (left) one character and deletes that character. When entering data, if a field becomes full, the cursor automatically advances to the next field. A carriage return may be entered to advance the cursor from an incomplete field.

5.0 SEARCH/BROWSE/EDIT DATA

When selection 3 is chosen from the Main Menu, the screen appears as in Figure 5. This module allows the user to browse the database on selected criteria and edit the database. The search criteria can be specific or a range may be given.

The specific search criteria combinations may be specified as follows:

SIC

SCC

Pollutant

SIC + SCC

SIC + SCC + Pollutant

SCC + Pollutant

For a retrieval of SIC + Pollutant, retrieve on SIC and use "Next record" and "Previous record" (described below) to scroll the data.

The range search criteria combinations may be specified as follows:

SIC to SIC

SIC + SCC to SIC

SIC + SCC + Pollutant to SIC

SIC + SCC to SIC + SCC

SIC + SCC to SCC

SCC to SCC

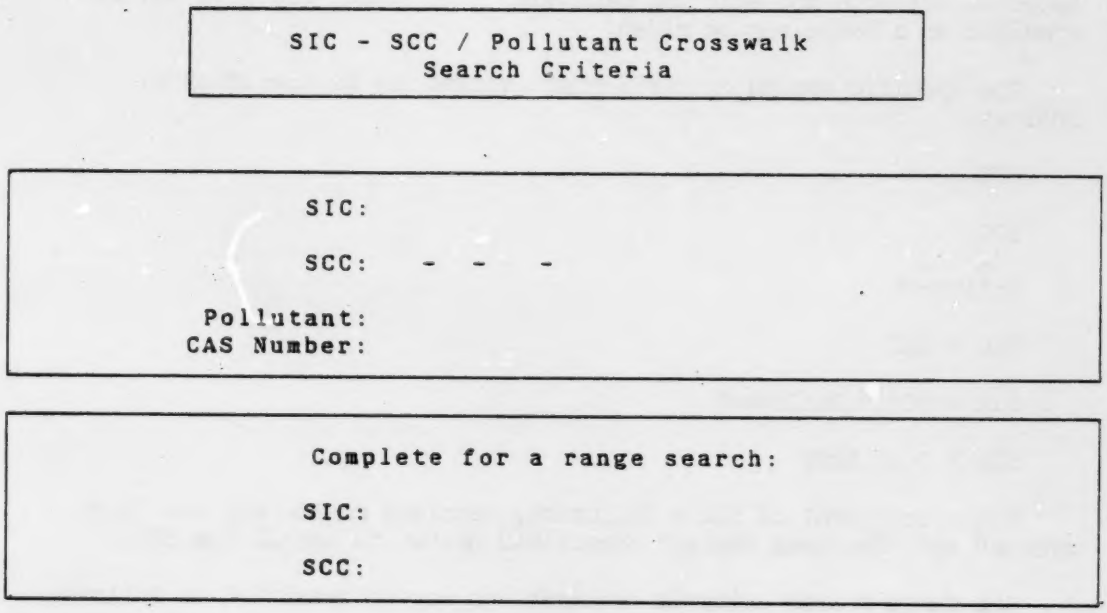
SCC + Pollutant to SCC

SIC + SCC + Pollutant to SCC

For all searches, search criteria must make an exact match for the first record found. However, a field may be partially filled as long as it is the last (or bottom-most) search specification. For example to do a range search for SIC numbers 2800 to 3000, 2800 must exist in the database. If the actual numbers in the database are 2802 to 2999 and the search criteria are 2800 to 3000, no matching data will be found; but if 2802 to 3000 are specified as the search criteria, the desired records will be found. All fields may be truncated during searches. For

Preceding page blank

FIGURE 5. SEARCH SCREEN



example, one, two, three, or four digits of a SIC code may be entered or a pollutant name may be shortened.

For a specific search, fill in the top data entry box on the screen with the desired field(s) for searching. The fields are examined in a top-down manner. For example when searching for SIC, SCC, and Pollutant, the SIC data field is searched first, then the SCC field, and the Pollutant field last. A message at the bottom of the screen will indicate if no match for the specified search criteria was found. To do a range search (having beginning and ending criteria), fill in the appropriate matching "to" field(s) in the lower data entry box.

Once completing the search criteria, the user may press Page Down to get the selections for the search at the bottom of the screen. These selections are "Quit" which returns to the main menu, "Redo screen" which places the cursor at the first field on the screen, and "search to Screen/Printer" in which the choice is made as to where the output of the search is to be routed. The valid selections for this screen would be Q, R, S or P, respectively.

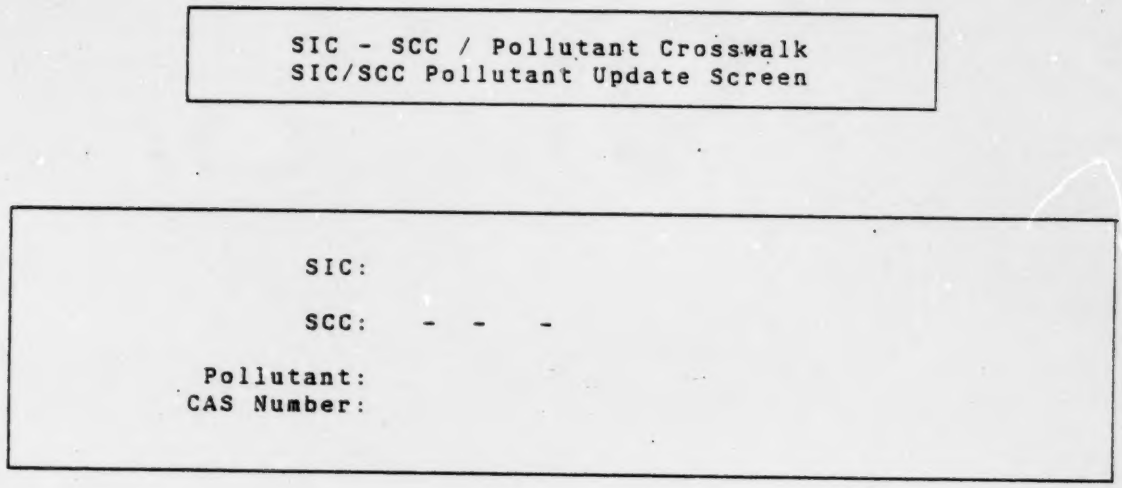
Upon selecting "search to Screen" the screen will appear as shown in Figure 6 if a match is found. The first record to match the criteria selected will appear in the data fields and several options will appear at the bottom of the screen. These options include "Edit" which will allow the user to edit the current record displayed on the screen, "Delete" which will delete a record from the database, "Next" for displaying the next record to meet the search criteria (if no more are found a message will appear at the bottom of the screen), "Previous" to skip backwards through selected records meeting the search criteria (if the record pointer is at the beginning of the selected criteria a message will be displayed at the bottom of the screen), and "Quit" which returns to the Search selection screen.

The Edit screen will appear as shown in Figure 7 and will allow the user to edit the displayed data. The options after editing the screen are "Quit/nosave" which returns to the browse screen, "Redo screen" which places the cursor at the first field on the screen, and "Save" which saves the current screen's data and returns to the browse mode.

FIGURE 6. BROWSE SCREEN

```
SIC - SCC / Pollutant Crosswalk
Browse Screen

SIC:

SCC:    -  -   -

Pollutant:
CAS Number:

Delete, Edit, Next record, Previous record or Quit:
```

FIGURE 7. EDIT SCREEN



6.0 SYNCHRONIZATION UTILITY

The Pollutant database serves as a reference table for the SIC/SCC/Pollutant database. These databases are associated by the CAS Number. When data is edited/updated in the Pollutant database (Selection 2 in the Main Menu), the data in the SIC/SCC/Pollutant database is not automatically updated. A utility for "synchronizing" the pollutant data in both files is necessary.

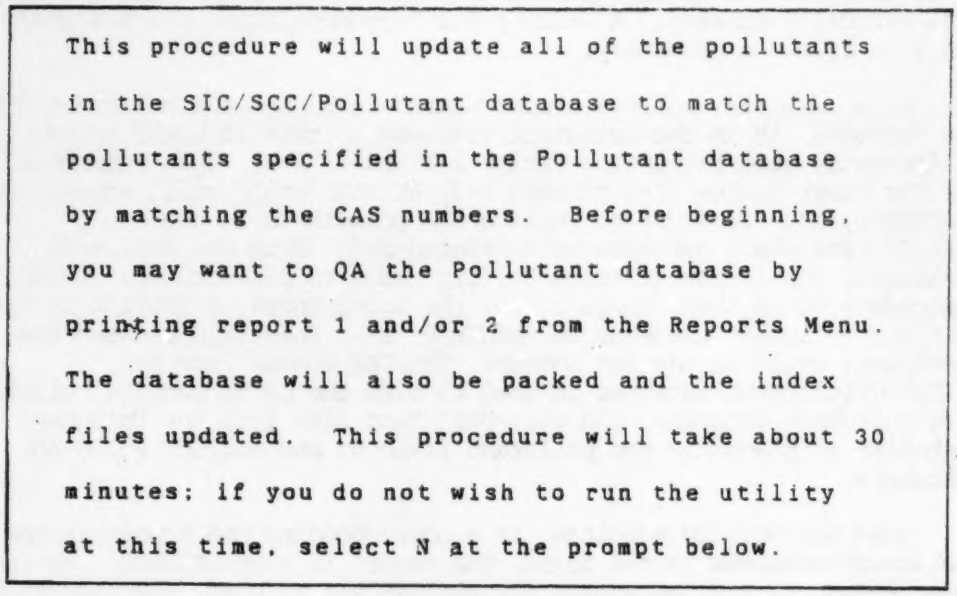
As an example, suppose a user wanted to change the pollutant BENZENE to "BENZENE, 1" in the Pollutant database so that it would appear differently upon reprinting Tables 2-1 and/or 2-2. Upon reprinting any of the other Tables (2-3 through 2-5) BENZENE would still appear, not "BENZENE, 1". In order to replace the pollutants in the SIC/SCC/Pollutant database with updated data from the Pollutant database, the synchronization utility needs to be executed. This procedure would then change all of the occurrences of BENZENE in the SIC/SCC/Pollutant database to "BENZENE, 1". The "key" between the databases would be the Cas Number. The CAS Number from the SIC/SCC/Pollutant database is used to find the corresponding pollutant in the Pollutant database, and this pollutant name from the Pollutant database is placed in the pollutant field of the SIC/SCC/Pollutant database.

When the utility executes, if a corresponding CAS Number in the Pollutant database is not found, the record is then DELETED. So in the case above, if the CAS Number for BENZENE was changed, the execution of this utility would DELETE all cases of BENZENE with the old CAS number in the SIC/SCC/Pollutant database. In this way, pollutants may also be deleted from both lists. First, the pollutant can be deleted from the Pollutant database with the Pollutant Update selection, and then it can be deleted from the SIC/SCC/Pollutant database with the synchronization utility (see Section 7.0 DELETE RECORDS for more information).

A screen as shown in Figure 8 will appear and the user will be prompted as whether to continue with the update or to return to the Main Menu. If the utility is selected to continue, a message at the bottom of the screen informs the user of the operation's progression. This procedure takes approximately 30 minutes on an IBM PC-AT machine.

Preceding page blank

FIGURE 8. SYNCHRONIZE SIC, SCC AND POLLUTANT DATA


This procedure will update all of the pollutants in the SIC/SCC/Pollutant database to match the pollutants specified in the Pollutant database by matching the CAS numbers. Before beginning, you may want to QA the Pollutant database by printing report 1 and/or 2 from the Reports Menu. The database will also be packed and the index files updated. This procedure will take about 30 minutes; if you do not wish to run the utility at this time, select N at the prompt below.
Continue (Y/N)?

7.0 DELETE RECORDS

To delete records from the SIC/SCC/Pollutant database, from the Search screen (Selection 4 from the Main Menu, Figure 6), select a criteria that will match the record selected for deletion. Once finding the record either by skipping forward or back with the "Previous" or "Next" commands (Figure 7) or by doing a specific search and matching directly, select "Delete" by typing a D from the Browse screen. A message will appear confirming the delete to be answered by either Y for "Yes" to delete or N for "No" to not delete. This query will prevent the accidental deletion of records from the database. A confirmed deleted record will be indicated by a message at the top of the screen in the Browse mode. The records will be physically deleted and the index files updated upon executing the Synchronization Utility.

To delete records from the Pollutant database, choose Selection 2 from the Main Menu and type in the desired pollutant. If the pollutant exists in the database, the pre-edit screen (Figure 4) will appear and the "Delete" choice may be selected by typing a D. A message confirming the delete will appear so pollutants will not accidentally be deleted from the database. The record will be marked for deletion as a message at the top of the screen will indicate. The physical deletion of the record will occur when the user returns to the Main Menu. Remember when changes are made to the Pollutant database, these changes are not automatically made to the SIC/SCC/Pollutant database, and the synchronization utility (Selection 3 from the Main Menu) must be executed to make the files match. In this case after deleting a pollutant from the Pollutant database and then running the utility, all records with CAS numbers in the SIC/SCC/Pollutant database not found in the Pollutant database will be deleted (in essence, all instances of the deleted pollutant from the Pollutant database found in the SIC/SCC/Pollutant database will be deleted).

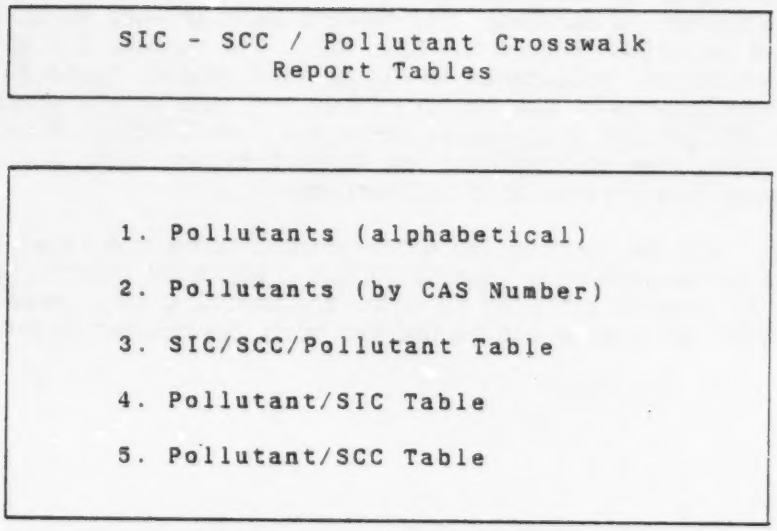
8.0 REPORTS

The specific reports for the SIC/SCC/Pollutant Crosswalk are Tables 2-1 through 2-5 contained in the accompanying crosswalk report. The reports are set up to automatically update the tables in the document after the database is updated. The Reports Menu appears as in Figure 9 after selecting number 5 from the Main Menu. The reports include Table 2-1, the listing of pollutants in alphabetical order; Table 2-2, the listing of pollutants by CAS number; Table 2-3, the cross-reference listing of SIC/SCC and pollutants; Table 2-4, the listing of pollutants and their associated SIC numbers; and Table 2-5, the listing of pollutants and their associated SCC numbers.

The user will be prompted to enter printer codes the first time any printing is to be generated (See SECTION 3.0 NOTES ON PRINTER OPERATION). The user will also be prompted to enter the starting page number for each report so that the tables can be updated with the correct page-numbering scheme.

Preceding page blank

FIGURE 9. REPORTS MENU



9.0 BACK UP FILES

Although it is not mandatory, it is recommended that the files be periodically backed up. This will save much time, effort, and headache if a hard disk or human error occurs which makes the files inaccessible, unusable or damaged. It is recommended that the DOS BACKUP command be used to backup the Crosswalk system and data files. The system with 10,000 database records fills approximately 3 high density diskettes with the DOS BACKUP command. Have at least 4 formatted high density diskettes ready before beginning. Also make sure the DOS BACKUP program is in the directory path (see Section 2.0 GETTING STARTED). From the C:\> prompt enter the command:

BACKUP C:\XWALK A:

The A drive on an IBM PC-AT is the high density drive and is the top or left drive depending on the computer's setup. The user will then be prompted to insert the diskettes in a sequential order, and the diskettes should be labeled with the number which the BACKUP program assigns. When the DOS prompt appears, the BACKUP procedure is complete.

If a hard disk failure or some other data damaging event occurs, it may become necessary to restore the backed up files. In order to restore the files from the backed up diskettes, make sure the DOS RESTORE program is in the directory path (see Section 2.0 GETTING STARTED). Make sure all of the files are deleted from the C:\XWALK directory. If the hard disk had to be reformatted, create the directory with the following command:

MKDIR \XWALK

Remember that the backup diskettes are only updated when the BACKUP program is executed for the XWALK directory. If it does become necessary to restore the files, the files will only be as up-to-date as the last time they were backed up. To restore the files, from the C:\> disk prompt enter the command:

CD\XWALK

From the C:\XWALK> disk prompt, enter the command:

RESTORE A: C:

The user will be prompted to enter the backed up diskettes in sequential order as they were labeled.

10.0 NOTES ON INDEX FILES

The index files can be recreated if necessary by using dBase III PLUS ®. The following example will demonstrate how to create the index SIC_SCC.NDX from the database SIC_SCC.DBF.

Under dBase III PLUS ® at the dot prompt, select the database to be indexed by using it as follows:

```
USE \XWALK\SIC_SCC
```

To create the index, enter the following command at the dot prompt:

```
INDEX ON <key> to \XWALK\<filename>
```

where <key> is the index key expression and <filename> is the name of the index file that will be created. The key index expression for each of the index files is shown in Figure 10. For this example the complete index expression is:

```
INDEX ON SIC + SCC + POLLUTANT TO \XWALK\SIC_SCC
```

The database and index file combinations can be found in Section 2.0 GETTING STARTED.

Preceding page blank

FIGURE 10. SIC/SCC/POLLUTANT CROSSWALK INDEX FILES

```
SIC_SCC.NDX
sic + scc + pollutant

SCC_POLL.NDX
scc + pollutant + sic

POLL_SIC.NDX
pollutant + sic + scc

POLL.NDX
pollutant

CAS.NDX
val(stuff((stuff(cas ^H_no,(at('-',cas ^H_no)),1,'')),(at('-',
   (stuff(cas ^H_no,(at('-',cas ^H_no)),1,'')))),1,''))

CAS1.NDX
cas_no
```

11.0 NOTES ON COMPILING THE SOURCE CODE

The source code used to compile the main program for the SIC/SCC/Pollutant Crosswalk - XWALK.EXE - is contained on the distribution diskettes. The following procedure describes how to recompile the source code if the need should arise. Only someone with experience using computers and preferably with compilers should attempt this procedure.

The QuickSilver® Compiler by WordTech Systems (version 1.1 or later) is necessary to compile the source code. First all of the source code program files with the extension PRG and format files with the extension FMT should be on the hard disk. Next certain files from the QuickSilver disks should be copied onto the hard disk. These files are:

```
DB3C.EXE
DB3L.EXE
DB3PCL.LIB
QS.EXE
QS.OVL
QS.LIB
QSPC1.LIB
QSPC2.LIB
```

These are all of the necessary files at the time this manual was printed. Subsequent versions of the compiler may change the name or number of these files. The best reference on the compilation of these files can be found in the QuickSilver® reference manual.

Finally the linker program - LINK.EXE - which comes with DOS must be in the directory path or in the same directory.

Once all of the files are together, an ascii file called FILES needs to be created with one line:

```
XWALK.PRG
```

The following commands will then compile the application:

```
DB3C -A -O @FILES
DB3L -Q @FILES
QS XWALK
LINK @XWALK.LNK
```

These commands must be entered sequentially at the DOS prompt or from a batch file. Several other files will be created with the extensions OBJ, DBC, and LNK. These files may be deleted upon successful compilation. The executable program will be XWALK.EXE.

APPENDIX A

SOURCE CODE LISTINGS

```
*xwalk.prg

set talk off
set bell off
set safety off
set confirm off
set scoreboard off
store space(1) to sel
clear
do while .not. sel$'NY'
   store space(1) to sel
   @ 24,0 say 'Are you using a color monitor (Y/N)?' get sel picture '!'
   read
enddo
if sel$'Y'
   set color to bg/n, w+/b, n
endif
do init
store .f. to del
store space(1) to sel
do while .not. sel$'12345Q'
   set format to xwalk
   read
   close format
   do case
      case sel = '1'
         do entry
      case sel = '2'
         do poll
      case sel = '3'
         set procedure to browse
         do browse
         close procedure
      case sel = '4'
         do synch
      case sel = '5'
         set procedure to misc
         do misc
         close procedure
      case sel = 'Q'
         exit
   endcase
   store space(1) to sel
enddo
```

```
*init.prg

@  3,9  to  4,14
@  8,64 to  9,69
store 1 to cntr
do while cntr < 10
   store (1+cntr) to cntr
enddo
clear
@  3,9  to  4,19
@  8,59 to  9,69
store 1 to cntr
do while cntr < 10
   store (1+cntr) to cntr
enddo
clear
@  3,9  to  5,19
@  7,59 to  9,69
store 1 to cntr
do while cntr < 10
   store (1+cntr) to cntr
enddo
clear
@  3,9  to  5,29
@  7,49 to  9,69
store 1 to cntr
do while cntr < 10
   store (1+cntr) to cntr
enddo
clear
@  3,9  to  6,29
@  6,49 to  9,69
store 1 to cntr
do while cntr < 10
   store (1+cntr) to cntr
enddo
clear
@  3,9  to  6,39
@  6,39 to  9,69
store 1 to cntr
do while cntr < 10
   store (1+cntr) to cntr
enddo
clear
@  3,9  to  7,39
@  5,39 to  9,69
store 1 to cntr
do while cntr < 10
   store (1+cntr) to cntr
enddo
```

```
clear
@  3,9  to  7,49
@  5,29 to  9,69
store 1 to cntr
do while cntr < 10
   store (1+cntr) to cntr
enddo
clear
@  3,9  to  8,49
@  4,29 to  9,69
store 1 to cntr
do while cntr < 10
   store (1+cntr) to cntr
enddo
clear
@  3,9  to  8,59
@  4,19 to  9,69
store 1 to cntr
do while cntr < 10
   store (1+cntr) to cntr
enddo
clear
@  3,9  to  9,59
@  3,19 to  9,69
store 1 to cntr
do while cntr < 10
   store (1+cntr) to cntr
enddo
clear
@  3,9  to  9,69
@  5,24 say 'SIC - SCC / Pollutant Crosswalk'
@  7,18 say 'U. S. EPA Air Management Technology Branch'
@ 11,24 to 17,54
@ 12,27 say '          '
@ 13,27 say '             '
@ 14,27 say '        '
@ 15,27 say ''
@ 16,27 say '          RTP, NC  1987'
store 1 to cntr
do while cntr <= 3
   time = time()
   do while time() = time
   enddo
   store (1+cntr) to cntr
enddo
return
```

```
*xwalk.fmt

@  3,24  say 'SIC - SCC / Pollutant Crosswalk'
@  4,35  say 'Main Menu'
@  2,18  to  5,61
@  9,23  say '1. Enter SIC/SCC/Pollutant data'
@ 11,23  say '2. Enter Pollutant data'
@ 13,23  say '3. Search/Browse/Edit data'
@ 15,23  say '4. Synchronization Utility'
@ 17,23  say '5. Reports'
@  7,18  to 19,61
@ 24,32  say '1 - 5 or Quit:' get sel picture '!'
```

```
*entry.prg

store space(1) to sel
store space(4)  to sic
store space(11) to scc
store space(15) to cas
store space(32) to pollutant
do while sel$' '
   set format to entry
   read
   close format
   use poll index poll, cas, cas1
   seek M->pollutant
   if found()
      store cas_no to cas
   else
      store space(15) to cas
   endif
   @ 16,30 get M->cas picture '!!!!!!!!!!!!!!!'
   read
   do while .not. sel$'AEQR'
      store space(1) to sel
      @ 24,15 say 'Add more, End/save, Quit/nosave or Redo screen:';
get sel picture '!'
      read
      @ 24,0 say space(80)
   enddo
   if sel$'AE'
      if M->scc = ' -  -   -  '
         store space(11) to scc
      endif
      @ 24,24 say 'Please wait, updating files...'
      use sic_scc index sic_scc, scc_poll, poll_sic
      append blank
      replace sic with M->sic, scc with M->scc, pollutant with :
M->pollutant, cas with M->cas
      use
      @ 24,0 say space(80)
   endif
   if sel$'EQ'
      use
      return
   endif
   store space(1) to sel
enddo
```

```
*entry.fmt

@  3,24 say 'SIC - SCC / Pollutant Crosswalk'
@  4,24 say 'SIC/SCC/Pollutant Update Screen'
@  2,18 to  5,61
@ 11,18 say '        SIC:' get M->sic       picture '9999'
@ 13,18 say '        SCC:' get M->scc       picture '9-99-999-99'
@ 15,18 say ' Pollutant:' get M->pollutant picture '!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!'
@ 16,18 say 'Cas Number:'
@  9, 8 to 18,71
```

```
*poll.prg

store .f. to pd
store space(1) to sel
store space(15) to cas
store space(32) to pollutant
use poll index poll, cas, cas1
do while sel$' '
   clear
   @  3,24 say 'SIC - SCC / Pollutant Crosswalk'
   @  4,28 say 'Pollutant Update Screen'
   @  2,18 to  5,61
   @ 12,18 say ' Pollutant:' get M->pollutant picture '!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!'
   @ 14,18 say 'Cas Number:'
   @ 10, 8 to 16,71
   read
   close format
   seek M->pollutant
   if found()
      store cas_no to cas
   else
      store space(15) to cas
   endif
   if found()
      store space(1) to sel
      @ 12,30 say M->pollutant
      @ 14,30 say M->cas
      if deleted()
         @  1, 0 say 'This record is marked for deletion'
      endif
      do while .not. sel$'DERQ'
         store space(1) to sel
         @ 24,24 say 'Delete, Edit, Retry or Quit:' get sel picture '!'
         read
         @ 24,0 say space(80)
      enddo
      if sel = 'D'
         store space(1) to sel
         do while .not. sel$'NY'
            store space(1) to sel
            @ 24,28 say 'Confirm Delete (Y/N)?' get sel picture '!'
            read
            @ 24,0 say space(80)
         enddo
         if sel$'Y'
            delete
            store .t. to pd
            @  1, 0 say 'This record is marked for deletion'
         endif
```

```
            store 'D' to sel
         endif
         if sel = 'E'
            @ 12,30 get M->pollutant picture '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'
            @ 14,30 get M->cas       picture '!!!!!!!!!!!!!!!'
            read
            store space(1) to sel
         endif
      else
         @ 14,30 get M->cas picture '!!!!!!!!!!!!!!!'
         read
      endif
      do while .not. sel$'AEQR'
         store space(1) to sel
         @ 24,12 say 'Add/edit more, End/save, Quit/nosave or Redo screen:' :
get sel picture '!'
         read
         @ 24,0 say space(80)
      enddo
      if sel$'AE'
         @ 24,24 say 'Please wait, updating files...'
         if .not. found()
            append blank
         endif
         replace pollutant with M->pollutant, cas_no with M->cas
         store space(15) to cas
         store space(32) to pollutant
         @ 24,0 say space(80)
      endif
      if sel$'EQ'
         if pd
            @ 24,24 say 'Please wait, updating files...'
            pack
            @ 24,0 say space(80)
         endif
         use
         return
      endif
      store space(1) to sel
   enddo
```

```
*browse.prg

PROCEDURE BROWSE
store space(1) to sel
do while .not. sel$'Q'
   if file ('browse.ndx')
      use
      erase browse.ndx
   endif
   if .not. sel$'R'
      store space(4)  to sic
      store space(4)  to sic1
      store space(11) to scc
      store space(11) to scc1
      store space(32) to pollutant
   else
      if M->sic <> space(4)
         store M->sic + space(len(M->sic) - 4) to sic
      endif
      if sic1 <> space(4)
         store sic1 + space(len(sic1) - 4) to sic1
      endif
      if len(M->scc) = 1
         store M->scc + '-  -    -  ' to scc
      endif
      if len(M->scc) = 3
         store M->scc + ' -    -  ' to scc
      endif
      if len(M->scc) = 4
         store M->scc + '-    -  ' to scc
      endif
      if len(M->scc) = 6
         store M->scc + '  -  ' to scc
      endif
      if len(M->scc) = 7
         store M->scc + ' -  ' to scc
      endif
      if len(M->scc) = 9
         store M->scc + '  ' to scc
      endif
      if len(M->scc) = 10
         store M->scc + ' ' to scc
      endif
      if len(scc1) = 1
         store scc1 + '-  -    -  ' to scc1
      endif
      if len(scc1) = 3
         store scc1 + ' -    -  ' to scc1
      endif
      if len(scc1) = 4
```

```
            store scc1 + '-    -   ' to scc1
         endif
         if len(scc1) = 6
            store scc1 + '  -  ' to scc1
         endif
         if len(scc1) = 7
            store scc1 + ' -  ' to scc1
         endif
         if len(scc1) = 9
            store scc1 + '  ' to scc1
         endif
         if len(scc1) = 10
            store scc1 + ' ' to scc1
         endif
         if M->pollutant <> space(32)
            store M->pollutant + space(len(M->pollutant) - 32) to polluant
         endif
      endif
      set format to browse
      read
      close format
      store space(1) to sel
      do while .not. sel$'QRSP'
         store space(1) to sel
         @ 24,17 say 'Quit, Redo screen or search to Screen/Printer:';
   get sel picture '!'
         read
         @ 24,0 say space(80)
      enddo
      if sel$'Q'
         use
         return
      endif
      if sel$'SP'
         @ 24,33 say 'Searching...'
         store space(1) to ndx, ndxspec, ndxspec1, ndxspec2
         if len(trim(M->sic)) <> 0
            store trim(M->sic) to ndxspec
            store ('sic') to ndx
            if len(trim(sic1)) <> 0
               store trim(sic1) to ndxspec1
               store 'sic' to ndxspec2
            else
               store trim(M->sic) to ndxspec1
               store 'sic' to ndxspec2
            endif
            use sic_scc index sic_scc, scc_poll, poll_sic
            seek ndxspec
            if substr(M->scc,1,1) <> ' '
               if substr(M->scc,8,1) <> ' '
```

```
      else
         if substr(M->scc,4,1) <> ' '
            store substr(M->scc,1,4) to scc
         else
            store substr(M->scc,1,1) to scc
         endif
      endif
      store M->sic-trim(M->scc) to ndxspec
      store (ndx + 'scc') to ndx
      if substr(scc1,1,1) <> ' '
         if substr(M->scc1,8,1) <> ' '
         else
            if substr(M->scc1,4,1) <> ' '
               store substr(M->scc1,1,4) to scc1
            else
               store substr(M->scc1,1,1) to scc1
            endif
         endif
         store ndxspec1 + trim(scc1) to ndxspec1
         store ndxspec2 + ' + scc' to ndxspec2
      else
         store ndxspec1 + trim(M->scc) to ndxspec1
         store ndxspec2 + ' + scc' to ndxspec2
      endif
      if len(trim(M->pollutant)) <> 0
         store M->sic + M->scc + trim(M->pollutant) to ndxspec
         store (ndx + 'pollutant') to ndx
         store ndxspec1 + trim(M->pollutant) to ndxspec1
         store ndxspec2 + ' + pollutant' to ndxspec2
      endif
   endif
   use sic_scc index sic_scc, scc_poll, poll_sic
   seek ndxspec
else
   if substr(M->scc,1,1) <> ' '
      if substr(M->scc,8,1) <> ' '
      else
         if substr(M->scc,4,1) <> ' '
            store substr(M->scc,1,4) to scc
         else
            store substr(M->scc,1,1) to scc
         endif
      endif
      store trim(M->scc) to ndxspec
      store ('scc') to ndx
      if substr(scc1,1,1) <> ' '
         if substr(M->scc1,8,1) <> ' '
         else
            if substr(M->scc1,4,1) <> ' '
               store substr(M->scc1,1,4) to scc1
```

```
                  else
                     store substr(M->scc1,1,1) to scc1
                  endif
               endif
               store trim(scc1) to ndxspec1
               store 'scc' to ndxspec2
            else
               store trim(M->scc) to ndxspec1
               store 'scc' to ndxspec2
            endif
            if len(trim(M->pollutant)) <> 0
               store M->scc + trim(M->pollutant) to ndxspec
               store (ndx + 'pollutant') to ndx
               store ndxspec1 + trim(M->pollutant) to ndxspec1
               store ndxspec2 + ' + pollutant' to ndxspec2
            endif
            use sic_scc index scc_poll, sic_scc, poll_sic
            seek ndxspec
         else
            if len(trim(M->pollutant)) <> 0
               store trim(M->pollutant) to ndxspec
               store ('pollutant') to ndx
               store trim(M->pollutant) to ndxspec1
               store 'pollutant' to ndxspec2
               use sic_scc index poll_sic, sic_scc, scc_poll
               seek ndxspec
            endif
         endif
      endif
      if eof()
         store space(1) to sel
         @ 24, 3 say 'No records found to meet the search criteria,:
press a key to continue...'
         set console off
         wait
         set console on
         @ 24,0 say space(80)
         store 'R' to sel
      endif
      if sel$'S'
         do scr
      endif
      if sel$'P'
         do lpt
         set print off
         set device to screen
         eject
         eject
      endif
   endif
```

```
enddo

PROCEDURE SCR
store space(1) to sel
do while .not. sel$'Q' .and. &ndxspec2 <= ndxspec1
   store space(1) to sel
   store sic to sic
   store scc to scc
   store pollutant to pollutant
   store cas to cas
   do browse1
   if deleted()
      @  0, 0 say 'This record is marked for deletion'
   endif
   store space(1) to sel
   do while .not. sel$'DENPQ'
      store space(1) to sel
      @ 24,14 say 'Delete, Edit, Next record, Previous record or Quit:';
get sel picture '!'
      read
      @ 24,0 say space(80)
   enddo
   if sel$'D'
      store space(1) to sel
      do while .not. sel$'YN'
         @ 24,28 say 'Confirm delete (Y/N)?' get sel picture '!'
         read
         @ 24, 0 say space(80)
      enddo
      if sel$'Y'
         delete
      endif
      store 'D' to sel
   endif
   if sel$'E'
      store space(1) to sel
      do while .not. sel$'QS'
         store space(1) to sel
         set format to entry
         read
         close format
         select 2
         use poll index poll
         seek M->pollutant
         if found()
            store cas_no to cas
         endif
         use
         @ 16,30 get M->cas picture '!!!!!!!!!!!!!!!'
         read
```

```
         select 1
         do while .not. sel$'QRS'
            @ 24,23 say 'Quit/nosave, Redo screen or Save:';
get sel picture '!'
            read
            @ 24,0 say space(80)
         enddo
         if sel$'S'
            if M->scc = ' -  -   -  '
               store space(11) to scc
            endif
            @ 24,24 say 'Please wait, updating files...'
            replace sic with M->sic, scc with M->scc, pollutant with :
M->pollutant, cas with M->cas
            @ 24,0 say space(80)
         endif
      enddo
      store 'E' to sel
   endif
   if sel$'N'
      store recno() to holdrec
      if .not. eof() .and. &ndxspec2 <= ndxspec1
         skip
      endif
      do while deleted()
         skip
      enddo
      if eof() .or. &ndxspec2 > ndxspec1
         @ 24, 2 say 'This is the last record to meet the criteria.;
press any key to continue...'
         set console off
         wait
         set console on
         @ 24,0 say space(80)
         go holdrec
      endif
   endif
   if sel$'P'
      store recno() to holdrec
      if .not. bof() .and. &ndxspec2 >= ndxspec
         skip -1
      endif
      do while deleted()
         skip -1
      enddo
      if bof() .or. &ndxspec2 < ndxspec
         @ 24, 2 say 'This is the first record to meet the criteria.;
press any key to continue...'
         set console off
         wait
```

```
            set console on
            @ 24,0 say space(80)
            go holdrec
         endif
      endif
      if sel$'Q'
         store space(1) to sel
         return
      endif
   enddo

   PROCEDURE BROWSE1
   clear
   @  3,24 say 'SIC - SCC / Pollutant Crosswalk'
   @  4,33 say 'Browse Screen'
   @  2,18 to  5,61
   @ 11,18 say '        SIC: '+ M->sic
   @ 13,18 say '        SCC: '+ M->scc
   @ 15,18 say ' Pollutant: '+ M->pollutant
   @ 16,18 say 'Cas Number: '+ M->cas
   @  9, 8 to 18,71
   return

   PROCEDURE LPT
   @ 24,0 say space(80)
   if .not. file ('prt.mem')
      store space(15) to prtcds
      do while prtcds = space(15)
         @ 23,12 say 'Enter the desired decimal print codes for your printer:'
         @ 24,32 get prtcds picture '999-999-999-999'
         read
      enddo
      @ 23,0 say space(80)
      @ 24,0 say space(80)
      store at(' ',prtcds) to termin
      store 1 to cntr
      if termin = 0
         store 15 to termin
      endif
      do while cntr < termin
         store substr(prtcds,cntr,3) to cd
         if cntr = 1
            store (chr(val(cd))) to reset
         else
            store (reset + chr(val(cd))) to reset
         endif
         store (cntr + 4) to cntr
      enddo
      save to prt all like reset
   else
```

```
   restore from prt additive
endif
@ 24,24 say 'Printing... Press <F10> to abort'
store space(3) to s3
store space(11) to s11
store 1 to pg
set print on
set device to print
@ 0,0 say reset
do while .not. eof()
   if inkey() = -9
      return
   endif
   if &ndxspec2 > ndxspec1
      exit
   endif
   store 7 to  lne
   @ lne,23 say 'SIC - SCC / Pollutant Crosswalk Search:
Page '+ltrim(str(pg,3,0))
   store (lne + 1) to lne
   store (pg + 1) to pg
   if ndxspec1 <> '9999'
      store len(ndx + ' = ' + ndxspec + ' to ' + ndxspec2:
+ ' = ' + ndxspec1) to x
      if x > 1
         store (39 - (x/2)) to x
         @ lne,x say (ndx + ' = ' + ndxspec + ' to '+:
ndxspec2 + ' = ' + ndxspec1)
         store (lne + 1) to lne
      endif
   else
      @ lne,32 say 'Entire listing'
   endif
   store (lne + 2) to lne
   @ lne,2 say ' SIC      SCC                          Pollutant:
                  CAS Number '
   store (lne + 1) to lne
   @ lne,2 say '----   -----------   -------------------------:
---------------   ---------------'
   store (lne + 1) to lne
   do while lne < 60 .and. &ndxspec2 <= ndxspec1
      @ lne,2 say sic+s3+scc+s3+pollutant+s11+cas
      store (lne + 1) to lne
      if .not. eof()
         skip
      endif
   enddo
enddo
return
```

```
*browse.fmt

@  3,24 say 'SIC - SCC / Pollutant Crosswalk'
@  4,32 say 'Search Criteria'
@  2,18 to  5,61
@  8,18 say 'Select field(s) to search on:'
@ 10,18 say '       SIC:' get M->sic       picture '9999'
@ 12,18 say '       SCC:' get M->scc       picture '9-99-999-99'
@ 14,18 say 'Pollutant:' get M->pollutant picture '!!!
```

```
*synch.prg

store space(1) to sel
do while .not. sel$'NY'
   store space(1) to sel
   set format to synch
   read
enddo
if sel$'N'
   return
endif
select 1
use sic_scc
select 2
use poll index cas1
select 1
store 1 to cntr
do while .not. eof()
  @ 24,22 say 'Updating record '+ltrim(str(cntr,5,0))+ ' out of '+;
ltrim(str(reccount(),5,0))
   store (cntr + 1) to cntr
   store .t. to fnd
   select 2
   seek sic_scc->cas
   if .not. found()
      store .f. to fnd
   endif
   select 1
   if fnd
      replace pollutant with poll->pollutant
   else
      delete
   endif
   skip
enddo
@ 24,0 say space(80)
@ 24,24 say 'Please wait, updating files...'
pack
run erase sic_scc.ndx
run erase scc_poll.ndx
run erase poll_sic.ndx
index on sic + scc + pollutant to sic_scc
index on scc + pollutant + sic to scc_poll
index on pollutant + sic + scc to poll_sic
@ 24,0 say space(80)
close all
return
```

```
*synch.fmt

@  3,16 say 'This procedure will update all of the pollutants'
@  5,16 say 'in the SIC/SCC/Pollutant database to match the'
@  7,16 say 'pollutants specified in the Pollutant database'
@  9,16 say 'by matching the CAS numbers.  Before beginning,'
@ 11,16 say 'you may want to QA the Pollutant database by'
@ 13,16 say 'printing report 1 and/or 2 from the Reports Menu.'
@ 15,16 say 'The database will also be packed and the index'
@ 17,16 say 'files updated.  This procedure will take about 30'
@ 19,16 say 'minutes; if you do not wish to run the utility at'
@ 21,16 say 'this time, select N at the prompt below.'
@  2,10 to 22,69
@ 24,31 say 'Continue (Y/N)?' get sel picture '!'
```

```
*misc.prg

PROCEDURE MISC
store space(1) to sel
do while .not. sel$ 'Q'
   store space(1) to sel
   set format to misc
   read
   close format
   @ 24,0 say space(80)
   if .not. file ('prt.mem')
      store space(15) to prtcds
      do while prtcds = space(15)
         @ 23,12 say 'Enter the desired decimal print codes for your printer:
         @ 24,32 get prtcds picture '999-999-999-999'
         read
      enddo
      @ 23,0 say space(80)
      @ 24,0 say space(80)
      store at(' ',prtcds) to termin
      store 1 to cntr
      if termin = 0
         store 15 to termin
      endif
      do while cntr < termin
         store substr(prtcds,cntr,3) to cd
         if cntr = 1
            store (chr(val(cd))) to reset
         else
            store (reset + chr(val(cd))) to reset
         endif
         store (cntr + 4) to cntr
      enddo
      save to prt all like reset
   else
      restore from prt additive
   endif
   if .not. sel$'Q'
      store 0 to pg
      @ 24,23 say 'Page numbering to starting with 2-'
      @ 24,57 get pg picture '999'
      read
      @ 24,0 say space(80)
      set print on
      set device to print
      @ 0,0 say reset
      set print off
      set device to screen
   endif
   do case
```

```
      case sel = '1'
         do rpt1
      case sel = '2'
         do rpt2
      case sel = '3'
         do rpt3
      case sel = '4'
         do rpt4
      case sel = '5'
         do rpt5
      case sel = 'Q'
         set escape off
         return
   endcase
   set console on
   set print off
   set device to screen
   eject
   eject
enddo

PROCEDURE RPT1
@ 24,24 say 'Printing... Press <F10> to abort'
set device to print
set print on
set console off
use poll index poll
store 5 to lne
@ lne,18 say 'Table 2-1.  LIST OF AIR TOXICS CONSIDERED'
store (lne + 1) to lne
@ lne,26 say '(Alphabetical by Pollutant)'
store (lne + 3) to lne
@ lne, 6 say '                POLLUTANT                          CAS Number'
store (lne + 1) to lne
@ lne, 6 say '----------------------------------------     --------------'
store (lne + 1) to lne
do while .not. eof()
   if inkey() = -9
      return
   endif
   if lne = 5
      @ lne,30 say 'Table 2-1., continued'
      store (lne + 3) to lne
      @ lne, 6 say '                POLLUTANT;
                        CAS Number'
      store (lne + 1) to lne
      @ lne, 6 say '------------------------;
----------------     --------------'
      store (lne + 1) to lne
   endif
```

```
      @ lne, 6 say pollutant
      @ lne,51 say cas_no
      store (lne + 1) to lne
      if .not. eof()
         skip
      endif
      if lne > 59 .or. eof()
         if pg < 10
            store 39 to x
         else
            store 38 to x
         endif
         @ 62,x say '2-'+ltrim(str(pg,3,0))
         store (pg + 1) to pg
         store 5 to lne
      endif
   enddo
   return

   PROCEDURE RPT2
   @ 24,24 say 'Printing... Press <F10> to abort'
   set device to print
   set print on
   set console off
   use poll index cas
   store 5 to lne
   @ lne,18 say 'Table 2-2.  LIST OF AIR TOXICS CONSIDERED'
   store (lne + 1) to lne
   @ lne,32 say '(By CAS Number)'
   store (lne + 3) to lne
   @ lne, 6 say '  CAS Number                          POLLUTANT'
   store (lne + 1) to lne
   @ lne, 6 say '---------------     ----------------------------------------'
   store (lne + 1) to lne
   do while .not. eof()
      if inkey() = -9
         return
      endif
      if lne = 5
         @ lne,30 say 'Table 2-2., continued'
         store (lne + 3) to lne
         @ lne, 6 say '  CAS Number                          POLLUTANT'
         store (lne + 1) to lne
         @ lne, 6 say '---------------     ------------------------;
   ----------------'
         store (lne + 1) to lne
      endif
      @ lne, 6 say cas_no
      @ lne,26 say pollutant
      store (lne + 1) to lne
```

```
   if .not. eof()
      skip
   endif
   if lne > 59 .or. eof()
      if pg < 10
         store 39 to x
      else
         store 38 to x
      endif
      @ 62,x say '2-'+ltrim(str(pg,3,0))
      store (pg + 1) to pg
      store 5 to lne
   endif
enddo
return

PROCEDURE RPT3
@ 24,24 say 'Printing... Press <F10> to abort'
set device to print
set print on
set console off
use sic_scc index sic_scc
store 5 to lne
@ lne,17 say 'Table 2-3.  SIC/SCC/Pollutant Crosswalk Index'
store (lne + 3) to lne
@ lne, 6 say ' SIC          SCC         POLLUTANT(S)'
store (lne + 1) to lne
@ lne, 6 say '----     -----------    ------------;
----------------------------------'
store (lne + 1) to lne
do while .not. eof()
   if inkey() = -9
      return
   endif
   store sic to key1
   store .f. to prt
   do while sic = key1
      if lne = 5
         @ lne,30 say 'Table 2-3., continued'
         store (lne + 3) to lne
         @ lne, 6 say ' SIC          SCC         POLLUTANT(S)'
         store (lne + 1) to lne
         @ lne, 6 say '----     -----------    ------------;
----------------------------------'
         store (lne + 1) to lne
         @ lne, 0 say ''
      endif
      if .not. prt
        @ lne, 6 say sic
         store .t. to prt
```

```
      endif
      store sic - scc to key2
      store .f. to prt1, prt2
      do while sic - scc = key2
         if scc = space(11)
            store '    . . .    ' to scc
         else
            store scc to scc
         endif
         if lne = 5
            @ lne,30 say 'Table 2-3., continued'
            store (lne + 3) to lne
            @ lne, 6 say ' SIC          SCC         POLLUTANT(S)'
            store (lne + 1) to lne
            @ lne, 6 say '----      -----------     ------------;
-------------------------------'
            store (lne + 1) to lne
            @ lne, 0 say ''
         endif
         if .not. prt1
            @ lne,15 say M->scc
            store .t. to prt1
         endif
         store len(trim(pollutant)) to x
         if (pcol() + x) > 73 .and. lne > 9
            store .f. to prt2
            ?? ';'
            store (lne + 1) to lne
         endif
         if .not. prt2 .and. lne < 60
            @ lne,30 say trim(pollutant)
            store .t. to prt2
         else
            if prt2 .and. lne < 60
               ?? '; ' + trim(pollutant)
            endif
         endif
         if lne < 60
            skip
         endif
         if lne > 59
            if pg < 10
               store 39 to x
            else
               store 38 to x
            endif
            @ 62,x say '2-'+ltrim(str(pg,3,0))
            store (pg + 1) to pg
            store 5 to lne
         endif
```

```
      enddo
      store (lne - 2) to lne
      if lne > 59 .or. eof()
         if pg < 10
            store 39 to x
         else
            store 38 to x
         endif
         @ 62,x say '2-'+ltrim(str(pg,3.0))
         store (pg + 1) to pg
         store 5 to lne
      endif
   enddo
enddo
return

PROCEDURE RPT4
@ 24,24 say 'Printing... Press <F10> to abort'
set device to print
set print on
set console off
use sic_scc index poll_sic
store 5 to lne
@ lne,22 say 'Table 2-4.  POLLUTANT/SIC Code Index'
store (lne + 3) to lne
@ lne, 6 say '          POLLUTANT (CAS Number)                    SIC Codes'
store (lne + 1) to lne
@ lne, 6 say '-----------------------------------------    ----------------:
---------'
store (lne + 1) to lne
do while .not. eof()
   if inkey() = -9
      return
   endif
   store pollutant to key1
   store .f. to prt
   store 50 to x
   do while pollutant = key1
      if lne = 5
         @ lne,30 say 'Table 2-4., continued'
         store (lne + 3) to lne
         @ lne, 6 say '          POLLUTANT (CAS Number):
                  SIC Codes'
         store (lne + 1) to lne
         @ lne, 6 say '-------------------------------:
---------    -------------------------'
         store (lne + 1) to lne
      endif
      if .not. prt
         @ lne, 6 say trim(pollutant) + ' (' + trim(cas) + ')'
```

```
            store .t. to prt
         endif
         @ lne,x say sic
         if x = 71
            store (lne + 1) to lne
         endif
         if x = 50
            store 57 to x
         else
            if x = 57
               store 64 to x
            else
               if x = 64
                  store 71 to x
               else
                  store 50 to x
               endif
            endif
         endif
         if lne > 59
            if pg < 10
               store 39 to x
             else
               store 38 to x
            endif
            @ 62,x say '2-'+ltrim(str(pg,3,0))
            store (pg + 1) to pg
            store 5 to lne
            store 50 to x
         endif
         store sic to sic
         do while pollutant = key1 .and. sic = M->sic .and. .not. eof()
            skip
         enddo
      enddo
      store (lne + 2) to lne
      if lne > 59 .or. eof()
         if pg < 10
            store 39 to x
         else
            store 38 to x
         endif
         @ 62,x say '2-'+ltrim(str(pg,3,0))
         store (pg + 1) to pg
         store 5 to lne
      endif
   enddo
   return

   PROCEDURE RPT5
```

```
@ 24,28 say 'Creating index file...'
use sic_scc
index on pollutant + scc + sic to poll_scc
@ 24,24 say 'Printing... Press <F10> to abort'
set device to print
set print on
set console off
store 5 to lne
@ lne,22 say 'Table 2-5.  POLLUTANT/SCC Code Index'
store (lne + 3) to lne
@ lne, 6 say '          POLLUTANT (CAS Number)                    SCC Codes'
store (lne + 1) to lne
@ lne, 6 say '-----------------------------------------    ----------------;
--------'
store (lne + 1) to lne
do while .not. eof()
   if inkey() = -9
      return
   endif
   store pollutant to key1
   store .f. to prt
   store 50 to x
   do while pollutant = key1
      if lne = 5
         @ lne,30 say 'Table 2-5., continued'
         store (lne + 3) to lne
         @ lne, 6 say '          POLLUTANT (CAS Number);
                   SCC Codes'
         store (lne + 1) to lne
         @ lne, 6 say '-------------------------------;
---------    ------------------------'
         store (lne + 1) to lne
      endif
      if .not. prt
         @ lne, 6 say trim(pollutant) + ' (' + trim(cas) + ')'
         store .t. to prt
      endif
      if scc <> space(11)
         @ lne,x say scc
         if x = 63
            store (lne + 1) to lne
         endif
         if x = 50
            store 63 to x
         else
            store 50 to x
         endif
      endif
      if lne > 59
         if pg < 10
```

```
            store 39 to x
         else
            store 38 to x
         endif
         @ 62,x say '2-'-ltrim(str(pg,3,0))
         store (pg + 1) to pg
         store 5 to lne
         store 50 to x
      endif
      store scc to scc
      do while pollutant = key1 .and. scc = M->scc .and. .not. eof()
         skip
      enddo
   enddo
   store (lne + 2) to lne
   if lne > 59 .or. eof()
      if pg < 10
         store 39 to x
      else
         store 38 to x
      endif
      @ 62,x say '2-'+ltrim(str(pg,3,0))
      store (pg + 1) to pg
      store 5 to lne
   endif
enddo
close all
erase poll_scc.ndx
return
```

```
*misc.fmt

@  3,24  say 'SIC - SCC / Pollutant Crosswalk'
@  4,33  say 'Report Tables'
@  2,18  to  5,61
@ 10,25  say '1. Pollutants (alphabetical)'
@ 12,25  say '2. Pollutants (by CAS Number)'
@ 14,25  say '3. SIC/SCC/Pollutant Table'
@ 16,25  say '4. Pollutant/SIC Table'
@ 18,25  say '5. Pollutant/SCC Table'
@  8,18  to 20,61
@ 24,32  say '1 - 5 or Quit:' get sel picture '!'
```